

[illegible]

```
EEEEEEEEEE RRRRRRRR RRRRRRRR SSSSSSSS UU UU BBBB BBBB 77777777 999999 000000
EEEEEEEEEE RRRRRRRR RRRRRRRR SSSSSSSS UU UU BBBB BBBB 77777777 999999 000000
EE          RR      RR RR      RR SS      UU      UU BB      BB 77 99 99 00 00
EE          RR      RR RR      RR SS      UU      UU BB      BB 77 99 99 00 00
EE          RR      RR RR      RR SS      UU      UU BB      BB 77 99 99 00 00
EE          RR      RR RR      RR SS      UU      UU BB      BB 77 99 99 00 00
EEEEEEEEEE RRRRRRRR RRRRRRRR SSSSSS  UU      UU BBBB BBBB 77 99999999 00 00 00
EEEEEEEEEE RRRRRRRR RRRRRRRR SSSSSS  UU      UU BBBB BBBB 77 99999999 00 00 00
EE          RR  RR  RR  RR  SS      UU      UU BB      BB 77 99 0000 00
EE          RR  RR  RR  RR  SS      UU      UU BB      BB 77 99 0000 00
EE          RR      RR RR      RR SS      UU      UU BB      BB 77 99 00 00
EE          RR      RR RR      RR SS      UU      UU BB      BB 77 99 00 00
EEEEEEEEEE RR      RR RR      RR SSSSSSSS UUUUUUUUU BBBB BBBB 77 999999 000000
EEEEEEEEEE RR      RR RR      RR SSSSSSSS UUUUUUUUU BBBB BBBB 77 999999 000000
                                     ....
                                     ....
                                     ....
                                     ....
```

```
LL          IIIIII SSSSSSSS
LL          IIIIII SSSSSSSS
LL          II     SS
LL          II     SS
LL          II     SS
LL          II     SS
LL          II     SSSSSS
LL          II     SSSSSS
LL          II     SS
LL          II     SS
LL          II     SS
LL          II     SS
LLLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLLL IIIIII SSSSSSSS
```

(4)	257	EX\$INIBOOTADP - INITIALIZE THE BOOT DEVICE ADAPTER
(5)	391	EX\$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
(5)	392	EX\$STARTUPADP - STARTUP ANY ADAPTERS
(6)	461	EX\$DUMPCPUREG - DUMP CPU-SPECIFIC IPR'S
(7)	577	EX\$READ TODR (P) - READ TIME-OF-DAY CLOCK
(8)	666	EX\$WRITE TODR (P) - WRITES TIME-OF-DAY CLOCK
(9)	724	EX\$REGSAVE - SAVE CPU-SPECIFIC IPR'S
(10)	786	EX\$REGRESTOR - RESTORE CPU-SPECIFIC IPR'S
(11)	846	EX\$INIPROCREG - CPU-DEPENDENT INITIALIZATION OF IPR'S
(12)	965	INISCACHE
(13)	985	SYSL\$CLRSBIA
(14)	1025	EX\$TEST_CSR
(15)	1197	ADPLINK = LINK ADAPTER CONTROL BLOCK INTO ADP LIST



```
0000 1      .NOSHOW CONDITIONALS
0000 5
0000 9
0000 13
0000 15      .TITLE  ERRSUB790 - ERROR SUBROUTINES FOR VAX 11/790
0000 17
0000 21
0000 22      .IDENT  'V04-002'
0000 23
0000 24
0000 25      *****
0000 26      *
0000 27      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 28      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 29      *  ALL RIGHTS RESERVED.
0000 30      *
0000 31      *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 32      *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 33      *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 34      *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 35      *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 36      *  TRANSFERRED.
0000 37      *
0000 38      *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 39      *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 40      *  CORPORATION.
0000 41      *
0000 42      *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 43      *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 44      *
0000 45      *
0000 46      *****
0000 47
0000 48      ++
0000 49
0000 50      FACILITY:
0000 51
0000 52      EXECUTIVE, LOADABLE SUBROUTINES USED BY POWERFAIL AND BUGCHECK.
0000 53
0000 54      ABSTRACT:
0000 55
0000 56      LOADABLE SUBROUTINES USED BY POWERFAIL AND BUGCHECK.
0000 57
0000 58      AUTHOR:
0000 59
0000 60      N. KRONENBERG, JULY 2, 1979.
0000 61
0000 62      MODIFIED BY:
0000 63
0000 64      V04-003 WMC00001      Wayne Cardoza      13-Sep-1984
0000 65      CRD reporting must not be turned off for VENUS.
0000 66
0000 67      V04-002 CWH4002      CW Hobbs      08-Sep-1984
0000 68      Correct typo in TCM0010, use '-' instead of '='
0000 69
0000 70      V04-001 TCM0010      Trudy C. Matthews      07-Sep-1984
0000 71      For the venus processor: move turning on cache from routine
```

```
0000 72 : EXE$INIPROCREG to a new routine: INISCACHE. Correct the
0000 73 : order in which registers are saved on the stack in EXE$REGSAVE.
0000 74 :
0000 75 : V03-022 TCM0009 Trudy C. Matthews 30-Jul-1984
0000 76 : When turning off CRD interrupts in EXE$INIPROCREG for VENUS,
0000 77 : read the processor register and write it back to preserve
0000 78 : the state of other bits in the register.
0000 79 :
0000 80 : V03-021 TCM0008 Trudy C. Matthews 23-Jul-1984
0000 81 : Remove venus code that queries the console for how to set up
0000 82 : cache and FBOX state. Instead always turn the cache and
0000 83 : FBOX on (and let the normal error handling code turn it off
0000 84 : if its bad).
0000 85 :
0000 86 : V03-020 DWT0214 David W. Thiel 02-May-1984
0000 87 : Revise MicroVAX I TODR register simulation.
0000 88 :
0000 89 : V03-019 KDM0096 Kathleen D. Morse 27-Mar-1984
0000 90 : Add missing indirection in MicroVAX I memory CSR
0000 91 : CRD enabling.
0000 92 :
0000 93 : V03-018 KPL0101 Peter Lieberwirth 4-Mar-1984
0000 94 : Add extra vectors now defined in SYSLOAVEC. These vectors
0000 95 : are insurance for v4.x
0000 96 :
0000 97 : V03-017 KPL0100 Peter Lieberwirth 12-Feb-1984
0000 98 : Change RPB$B_BOOTNDT to RPB$W_BOOTNDT, since BI devices
0000 99 : will have 16-bit device types.
0000 100 :
0000 101 : V03-016 KDM0092 Kathleen D. Morse 23-Jan-1984
0000 102 : Correct the number of cpu-specific IPRs logged for the
0000 103 : 11/730 and MicroVAX I cpus.
0000 104 :
0000 105 : V03-015 CWH8001 CW Hobbs 5-Dec-1983
0000 106 : Add entry points for EXE$READP_TODR and EXE$WRITEP_TODR
0000 107 : to access physical TODR register for Nautilus CPU. For
0000 108 : other processors, these amount to duplicate labels on
0000 109 : EXE$READ_TODR and EXE$WRITE_TODR.
0000 110 :
0000 111 : V03-014 KTA3088 Kerbey T. Altmann 17-Oct-1983
0000 112 : Fix bug in 730 conditional for EXE$INIBOOTADP.
0000 113 :
0000 114 : V03-013 KDM0081 Kathleen D. Morse 13-Sep-1983
0000 115 : Create Micro-VAX I version.
0000 116 :
0000 117 : V03-012 KDM0055 Kathleen D. Morse 12-Jul-1983
0000 118 : Move IPR PME into the cpu-dependent register save and
0000 119 : restore routines.
0000 120 :
0000 121 : V03-011 KDM0049 Kathleen D. Morse 07-Jul-1983
0000 122 : Add the following processor registers to the cpu-specific
0000 123 : dump IPRs routine: ICR, TODR, ACCS. Add usage of
0000 124 : register: EXE$READ_TODR and EXE$WRITE_TODR.
0000 125 :
0000 126 : V03-010 KDM0048 Kathleen D. Morse 07-Jul-1983
0000 127 : Add loadable routines for referencing the time-of-day
0000 128 : clock: EXE$READ_TODR, EXE$WRITE_TODR.
```



0000	129	:	
0000	130	:	
0000	131	:	V03-009 TCM0007 Trudy C. Matthews 02-Jun-1983
0000	132	:	Fix routine SYSLSCLRSBIA so that it calculates the address
0000	133	:	of SBI adapter register space correctly.
0000	134	:	
0000	135	:	V03-008 TCM0006 Trudy C. Matthews 9-Feb-1983
0000	136	:	Store enable/disable state of 11/790 cache and FBOX in
0000	137	:	EXESGB_CPUDATA cell during system initialization.
0000	138	:	
0000	139	:	V03-007 TCM0005 Trudy C. Matthews 11-Jan-1983
0000	140	:	Add routine SYSLSCLRSBIA. Add SBIA register initialization
0000	141	:	to EXESINIPROCREG. Add 11/790 machine check handler to
0000	142	:	EXESTEST CSR. Change 11/780 machine check handler to
0000	143	:	write PR\$ SBIFS back to itself to clear error bits.
0000	144	:	Add labels for two "extra" routines, that can be patched
0000	145	:	if extra vectors from SYS to SYSLOA are needed in between
0000	146	:	major releases. Make EXESDUMPCPUREG log the SBI registers
0000	147	:	from the SBI the 11/790 system disk is on.
0000	148	:	
0000	149	:	V03-006 TCM0004 Trudy C. Matthews 3-Jan-1983
0000	150	:	Add more 11/790-specific code.
0000	151	:	
0000	152	:	V03-005 TCM0003 Trudy C. Matthews 17-Dec-1982
0000	153	:	Add conditional assembly switch to the invocations
0000	154	:	of 11/790-specific definition macros.
0000	155	:	
0000	156	:	V03-004 TCM0002 Trudy C. Matthews 15-Dec-1982
0000	157	:	Added 11/790-specific code to EXESINIPROCREG.
0000	158	:	
0000	159	:	V03-003 TCM0001 Trudy C. Matthews 13-Dec-1982
0000	160	:	Added 11/790-specific code to power down/power up
0000	161	:	routines.
0000	162	:	
0000	163	:	V03-002 KTA3018 Kerbey T. Altmann 30-Oct-1982
0000	164	:	Remove CI and UBA routines to another module.
0000	165	:	--

```
0000 167
0000 168 :
0000 169 : MACRO LIBRARY CALLS:
0000 170 :
0000 171
0000 172 $ADPDEF ;DEFINE ADAPTER OFFSETS
0000 173 $BQODEF ;DEFINE BOOT QIO OFFSETS
0000 174 $BTODEF ;DEFINE BOOT DEVICE TYPES
0000 175 $EMBCRDEF ;DEFINE ERROR MSG BUFFER OFFSETS
0000 176 $IDBDEF ;DEFINE INTERRUPT DISPATCH OFFSETS
0000 177 $IPLDEF ;DEFINE INTERRUPT PRIORITY LEVELS
0000 178 $MBADEF ;DEFINE MASSBUS ADAPTER OFFSETS
0000 179 $NDTDEF ;DEFINE NEXUS DEVICE TYPES
0000 180 $PRDEF ;DEFINE INTERNAL PROCESSOR REGISTERS
0000 181 $RPBDEF ;DEFINE RESTART PARAM BLOCK OFFSETS
0000 182 $SSDEF ;DEFINE SYSTEM STATUS CODES
0000 183 $SUBDEF ;DEFINE UNIBUS ADAPTER OFFSETS
0000 185 $ACCSDEF ;DEFINE 11/790 FBOX STATUS REGISTER
0000 186 $CSWPDEF ;DEFINE 11/790 CACHE SWEEP REGISTER
0000 187 $EHSRDEF ;DEFINE 11/790 ERROR HANDLING STATUS REG
0000 188 $IO790DEF ;DEFINE 11/790 I/O ADDRESS SPACE
0000 189 $MCF790DEF ;DEFINE 11/790 MACHINE CHECK STACK FRAME
0000 190 $MERGDEF ;DEFINE 11/790 MEMORY ERROR REGISTER
0000 191 $MSTAT2DEF ;DEFINE 11/790 MEMORY STATUS REGISTER
0000 192 $PR790DEF ;DEFINE 11/790 INTERNAL PROCESSOR REGS
0000 193 $SBIADef ;DEFINE 11/790 SBIA REGISTERS
0000 195
0000 199
0000 203
0000 207
0000 211 :
0000 212 : EQUATED SYMBOLS:
0000 213 :
0000 218
0000 223
0000 228
00000001 0000 230 C780_LIKE = 1
00000000 0000 231 C750_LIKE = 0
0000 233
0000 238
0000 239 :
0000 240 : Define labels for two "extra" routines. This reserves some vectors from
0000 241 : SYS.EXE into SYSLOAxxx.EXE that can be patched if another routine must
0000 242 : be added in between major releases.
0000 243 :
0000 244 EXE$EXTRA1:: ; aligned
0000 245 EXE$EXTRA2:: ; aligned
0000 246 EXE$EXTRA3:: ; aligned
0000 247 EXE$EXTRA4:: ; aligned
0000 248 EXE$EXTRA5:: ; aligned
0000 249 EXE$EXTRA6:: ; packed
0000 250 EXE$EXTRA7:: ; packed
0000 251 EXE$EXTRA8:: ; packed
0000 252 EXE$EXTRA9:: ; packed
0000 253 EXE$EXTRA10:: ; packed (think this is enough?)
00 0000 254
0000 255 HALT ; Error if these labels are used.
```



```
0001 257 .SBTTL EXESINIBOOTADP - INITIALIZE THE BOOT DEVICE ADAPTER
0001 258
0001 259 :+ EXESINIBOOTADP - GET THE SYSTEM BOOT DEVICE ADAPTER AND INIT IT.
0001 260 : THIS ROUTINE IS CALLED FROM BUGCHECK BEFORE THE BOOTDRIVER IS CALLED.
0001 261
0001 262 : INPUTS:
0001 263
0001 264 : R6 = RPB ADDRESS
0001 265
0001 266 : OUTPUTS:
0001 267
0001 268 : R0-R2 DESTROYED
0001 269 : OTHER REGISTERS PRESERVED
0001 270 :-
0001 271
0000 272 .PSECT SYSLOA, LONG
0000 273 .ENABLE LSB
0000 274
0000 275 EXESINIBOOTADP:: :SUBROUTINE ENTRY
0000 276
0000 277
66 A6 91 0000 278 CMPB RPB$B_DEVTYPE(R6),- :IS BOOT DEVICE THE CONSOLE
40 8F 0003 279 #BTD$R_CONSOLE :BLOCK STORAGE DEVICE?
67 13 0005 280 BEQL 40$ :YES, RETURN
50 60 A6 D0 0007 281 MOVL RPB$L_ADVPTR(R6),R0 :GET ADDR OF ADAPTER REG SPACE
0008 282
0008 283
52 00A1 C6 03 AB 0008 284 BICW3 #3,RPB$W_BOOTNDT(R6),R2 :GET GENERIC ADAPTER TYPE
38 52 B1 0011 285 CMPW R2,#NDT$_CI :CI ADAPTER?
21 13 0014 286 BEQL 20$ :YES, RETURN
20 52 B1 0016 287 CMPW R2,#NDT$_MB :MASS BUS ADAPTER?
1D 12 0019 288 BNEQ INI_UBADP :BRANCH IF NOT
02 D0 001B 289 MOVL #MBT$M_CR_ABORT,- :ABORT ACTIVE TRANSFER
04 A0 001D 290 MBASL_CR(R0)
001F 291
001F 292
001F 293
001F 300
51 51 1B DB 001F 302 MFPR #PR790$_TODR,R1 :GET CURRENT TIME (10 MS UNITS)
51 64 A1 9E 0022 304 MOVAB 100(R1),R1 :ALLOW ONE SECOND
08 A0 D5 0026 305 10$: TSTL MBASL_SR(R0) :WAIT UNTIL TRANSFER
08 18 0029 306 BGEQ 15$ :IS COMPLETE
002B 307
002B 311
002B 315
52 1B DB 002B 317 MFPR #PR790$_TODR,R2 :GET CURRENT TIME
002E 319
52 51 D1 002E 320 CMPL R1,R2 :CHECK FOR INTERVAL EXPIRED
F3 1A 0031 321 BGTRU 10$ :NOT YET, WAIT SOME MORE
01 D0 0033 322 15$: MOVL #MBAS$M_CR_INIT,- :NOW INIT MBA
04 A0 0035 323 MBASL_CR(R0)
05 0037 324 20$: RSB :DONE
0038 325
0038 327
0038 331
0038 332 INI_UBADP: :INIT UBA
0038 333
0038 335
04 01 D0 0038 335 MOVL #UBAS$M_CR_INIT,- :INIT UBA
04 A0 003A 336 UBASL_CR(R0)
```



```
00010000 8F D3 003C 337 25$: BITL #UBASH_CSR_UBIC,-
60 0042 338 ;WAIT FOR UBA INIT
F7 13 0043 339 BEQL UBASH_CSR(R0) ; TO COMPLETE
0045 341
0045 358
0045 360
0045 361 : CHECK THE VMB VERSION NUMBER. IF IT EXISTS AND IF IT IS 7 OR GREATER, THEN
0045 362 : SEE IF ANY UNIBUS MAP REGISTERS TO DISABLE.
0045 363
0045 364
52 34 A6 D0 0045 365 MOVL RPB$$_IOVEC(R6),R2 ;PICK UP THE IOVECTOR FROM RPB
51 10 A2 B2 0049 366 MCOMW BQO$$_VERSION(R2),R1 ;GET VMB VERSION NUMBER 1'S COMPLEMENTED
12 A2 51 B1 004D 367 CMPW R1,BQO$$_VERSION+2(R2) ;CHECK AGAINST CHECK WORD IN VMB
18 12 0051 368 BNEQ 40$ ;IF NOT, ASSUME NO VERSION NUMBER
07 10 A2 B1 0053 369 CMPW BQO$$_VERSION(R2),#7 ;VERSION 7 OR GREATER OF VMB?
15 1F 0057 370 BLSSU 40$ ;NO, DON'T BOTH WITH UMR'S
52 24 A2 D0 0059 371 MOVL BQO$$_UMR_DIS(R2),R2 ;GRAB THE NUMBER OF UMR'S TO DISABLE
OF 13 005D 372 BEQL 40$ ;NONE, LEAVE
005F 373
04 A0 52 16 78 005F 375 ASHL #22,R2,UBASH_CR(R0) ;SET THE UMR DISABLE BITS
0064 377
0064 378 :
0064 379 : THIS CODE IS EXECUTED FOR ALL PROCESSORS. ITS DISABLES ANY UNIBUS MAP
0064 380 : REGISTERS ASSOCIATED WITH UNIBUS MEMORY TO PREVENT CONTENTION BETWEEN
0064 381 : SBI AND UNIBUS ADDRESSES.
0064 382 :
0064 383
51 0800 C0 DE 0064 384 MOVAL UBASH_MAP(R0),R1 ;ADDRESS OF FIRST REGISTER
81 D4 0069 385 30$: CLRL (R1)+ ;DISABLE IT
FB 52 F5 006B 386 SOBGTR R2,30$ ;LOOP UNTIL ALL DONE
05 006E 388 40$: RSB ;DONE WITH UBA INIT
006F 389 .DISABLE LSB
```

```
006F 391 .SBTTL EXESSHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
006F 392 .SBTTL EXESSSTARTUPADP - STARTUP ANY ADAPTERS
006F 393 :+
006F 394 : EXESSHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
006F 395 : THIS ROUTINE IS CALLED FROM BUGCHECK BEFORE THE DUMP IS TAKEN TO
006F 396 : ENSURE THAT ALL ADAPTERS THAT NEED TO BE QUIESENT ARE.
006F 397 :
006F 398 : INPUTS:
006F 399 :
006F 400 : IPL = 31
006F 401 :
006F 402 : OUTPUTS:
006F 403 :
006F 404 : OTHER REGISTERS PRESERVED
006F 405 :-
006F 406 : .ENABLE LSB
006F 407 :
006F 408 EXESSSTARTUPADP::
006F 409 PUSHF #M<R0,R1,R2,R4> ; Save a register
51 B6'AF BB 0071 410 MOVAL B^ADP_TBL_UP,R1 ; Address of startup table
06 11 0075 411 BRB 5$ ; Join common code
0077 412 :
0077 413 EXESSHUTDWNADP::
0077 414 PUSHF #M<R0,R1,R2,R4> ; Save a register
51 9E'AF BB 0079 415 MOVAL B^ADP_TBL_DWN,R1 ; Address of shutdown table
FFFFF9F DE 007D 416 5$: MOVAL @#<IOCSGL_ADPLIST- -
52 04 A2 DO 0083 417 ADPSL [LINK],R2 ; Get pointer to head of adapter list
11 13 0084 418 10$: MOVL ADPSL_LINK(R2),R2 ; Flink onward
54 62 DO 0088 419 BEQL 20$ ; Branch if at end of list
50 0E A2 3C 008A 420 MOVL ADPSL_CSR(R2),R4 ; Get address of CSR
50 6140 DE 008D 421 MOVZWL ADPSW_ADPTYPE(R2),R0 ; Get adapter type code
00 B040 16 0091 422 MOVAL (R1)[R0],R0 ; Get table entry of adap shutdown
E9 11 0095 423 JSB @<R0>[R0] ; Call adapter shutdown
0099 424 BRB 10$ ; Next adapter
009B 425 :
009B 426 20$: POPF #M<R0,R1,R2,R4>
009D 427 30$: RSB
009E 428 :
009E 429 :
009E 430 : Table of addresses of adapter shutdown routines ordered
009E 431 : by adapter type in ADPSW_ADPTYPE.
009E 432 :
009E 433 :
009E 434 ADP_TBL_DWN: ; Address table start
FFFFF7F 009E 435 .LONG 30$- ; 0-MBA
FFFFF7B 00A2 439 .LONG 30$- ; 1-UBA
FFFFF77 00A6 441 .LONG 30$- ; 2-DR32
FFFFF73 00AA 442 .LONG 30$- ; 3-MA780
FFFFF52' 00AE 443 .LONG C1$SHUTDOWN- ; 4-CI
FFFFFEB 00B2 444 .LONG 30$- ; Rsvrd for future expansion
00B6 445 :
00B6 446 :
00B6 447 : Table of addresses of adapter startup routines ordered
00B6 448 : by adapter type in ADPSW_ADPTYPE.
00B6 449 :
00B6 450 :
00B6 451 ADP_TBL_UP: ; Address table start
```

FFFFFF4A'	00B6	452	.LONG	MBASINITIAL-	:	0-MBA
FFFFFF46'	00BA	453	.LONG	UBASINITIAL-	:	1-UBA
FFFFFFDF	00BE	454	.LONG	30\$-	:	2-DR32
FFFFFF3E'	00C2	455	.LONG	MASINITIAL-	:	3-MA780
FFFFFFD7	00C6	456	.LONG	30\$-	:	4-CI
FFFFFFD3	00CA	457	.LONG	30\$-	:	Rsvrd for future expansion
	00CE	458				
	00CE	459	.DISABLE	LSB		



```
00CE 461      .SBTTL EXESDUMPCPUREG - DUMP CPU-SPECIFIC IPR'S
00CE 462      :+
00CE 463      : DUMP CPU-SPECIFIC IPR'S INTO ERROR MESSAGE BUFFER.
00CE 464      :
00CE 465      : TWENTY-FOUR LONGWORDS ARE RESERVED IN THE EMB FOR CPU-SPECIFIC
00CE 466      : IPR'S. THE FORMATS FOR VARIOUS CPU'S ARE:
00CE 467      :
00CE 468      : 11/780:      11/750:      11/730:      11/790:      UVAX I:
00CE 469      :
00CE 470      : ICR          ICR          ICR          ICR          UNUSED(0)
00CE 471      : TODR        TODR        TODR        TODR        APPROX TODR
00CE 472      : ACCS        ACCS        ACCS        ACCS        UNUSED(0)
00CE 473      : SBIFS      TBDR          21 UNUSED(0)  SBISTS (1st SBI)  21 UNUSED(0)
00CE 474      : SBISC      CADR          SILOCMP      :
00CE 475      : SBIMT      MCESR        MAINT        :
00CE 476      : SBIER      CAER          SBIERR       :
00CE 477      : SBIS      CMERR        TMOADDRS      :
00CE 478      : 16 SBI SILO 16 UNUSED(0) 16 SBI SILO :
00CE 479      :
00CE 480      : INPUTS:
00CE 481      :
00CE 482      : RO - ADDR IN EMB OF START OF CPU-SPECIFIC REGISTERS=
00CE 483      : OFFSET EMB$L_CR_CPUREG
00CE 484      :
00CE 485      : OUTPUTS:
00CE 486      :
00CE 487      : RO,R1 DESTROYED
00CE 488      : ALL OTHER REGISTERS PRESERVED
00CE 489      : -
00CE 490      :
00CE 491      : .ENABL LSB
00CE 492      :
00CE 493      : EXESDUMPCPUREG::      : SUBROUTINE ENTRY
00CE 494      :
00CE 495      :
00CE 509      :
00CE 510      :
00CE 524      :
00CE 525      :
00CE 536      :
00CE 537      :
00CE 539      : MFPR      #PR790$I CR,(R0)+      : LOG INTERVAL COUNT REG,
00CE 540      : MFPR      #PR790$TODR,(R0)+      : TIME-OF-DAY REG,
00CE 541      : MFPR      #PR790$ACCS,(R0)+      : ACCELERATOR CONTROL REG.
00CE 542      : MOVL      G^EXESGC RPB,R1      : GET ADDRESS OF RPB
00CE 543      : MOVL      RPB$L BOOTR1(R1),R1      : GET R1 INPUT TO VMB
00CE 544      : EXTZV      #RPB$V_ABUS,#RPB$S_ABUS,-: GET ABUS SLOT NUMBER OF SBI THAT
00CE 545      : R1,R1      : WE BOOTED FROM
00CE 546      : MOVL      W^ABUS_VA[R1],R1      : GET VIRTUAL ADDRESS OF SBI REGISTERS
00CE 547      : BEQL      90$      : BRANCH IF SBI NOT MAPPED
00CE 548      : MOVL      SBIASL_SBISTS(R1),(R0)+      : SBI FAULT STATUS REGISTER
00CE 549      : MOVL      SBIASL_SILOCMP(R1),(R0)+      : SBI SILO COMPARATOR
00CE 550      : MOVL      SBIASL_MAINT(R1),(R0)+      : SBI MAINTENANCE REGISTER
00CE 551      : MOVL      SBIASL_SBIERR(R1),(R0)+      : SBI ERROR REGISTER
00CE 552      : MOVL      SBIASL_TMOADDRS(R1),(R0)+      : SBI TIMEOUT ADDRESS
00CE 553      : MOVL      #16,-(SP)      : GET NUMBER OF SILO ENTRIES TO DUMP
00CE 554      : MOVL      SBIASL_SBI SILO(R1),(R0)+      : READ THE SILO 16 TIMES

51      80      1A      DB      00CE 539      MFPR      #PR790$I CR,(R0)+      : LOG INTERVAL COUNT REG,
      80      1B      DB      00D1 540      MFPR      #PR790$TODR,(R0)+      : TIME-OF-DAY REG,
      80      28      DB      00D4 541      MFPR      #PR790$ACCS,(R0)+      : ACCELERATOR CONTROL REG.
51      00000000'GF      DO      00D7 542      MOVL      G^EXESGC RPB,R1      : GET ADDRESS OF RPB
      51      20      A1      DO      00DE 543      MOVL      RPB$L BOOTR1(R1),R1      : GET R1 INPUT TO VMB
      02      04      EF      00E2 544      EXTZV      #RPB$V_ABUS,#RPB$S_ABUS,-: GET ABUS SLOT NUMBER OF SBI THAT
      51      51      51      00E3 545      R1,R1      : WE BOOTED FROM
51      0000'CF41      DO      00E7 546      MOVL      W^ABUS_VA[R1],R1      : GET VIRTUAL ADDRESS OF SBI REGISTERS
      20      13      00ED 547      BEQL      90$      : BRANCH IF SBI NOT MAPPED
      80      3C      A1      DO      00EF 548      MOVL      SBIASL_SBISTS(R1),(R0)+      : SBI FAULT STATUS REGISTER
      80      40      A1      DO      00F3 549      MOVL      SBIASL_SILOCMP(R1),(R0)+      : SBI SILO COMPARATOR
      80      44      A1      DO      00F7 550      MOVL      SBIASL_MAINT(R1),(R0)+      : SBI MAINTENANCE REGISTER
      80      34      A1      DO      00FB 551      MOVL      SBIASL_SBIERR(R1),(R0)+      : SBI ERROR REGISTER
      80      38      A1      DO      00FF 552      MOVL      SBIASL_TMOADDRS(R1),(R0)+      : SBI TIMEOUT ADDRESS
      7E      10      DO      0103 553      MOVL      #16,-(SP)      : GET NUMBER OF SILO ENTRIES TO DUMP
      80      30      A1      DO      0106 554      MOVL      SBIASL_SBI SILO(R1),(R0)+      : READ THE SILO 16 TIMES
```

F9 6E	F5	010A	555	SOBGTR (SP),10\$	: LOOP
8E	D5	010D	556	TSTL (SP)+	: POP TEMPORARY FROM STACK
		010F	558		
		010F	559		
		010F	572	90\$:	
	05	010F	573	RSB	
		0110	574	.DISABLE LSB	
		0110	575		





```
0114 666 .SBTTL EXESWRITE_TODR (P) - WRITES TIME-OF-DAY CLOCK
0114 667 :+
0114 668 : WRITES THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
0114 669 : DIFFERENT WAYS: AS AN INTERNAL PROCESSOR REGISTER, AS PART
0114 670 : OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE. IT
0114 671 : MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
0114 672 :
0114 673 : INPUTS:
0114 674 :
0114 675 : RO - CONTAINS VALUE TO BE WRITTEN INTO TODR
0114 676 :
0114 677 : OUTPUTS:
0114 678 :
0114 679 : NEW TIME VALUE WRITTEN INTO TODR.
0114 680 : ALL REGISTERS PRESERVED.
0114 681 :-
0114 682
0114 683 EXESWRITEP_TODR:: ; SUBROUTINE ENTRY
0114 684
0114 685 : NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
0114 686 : REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS. FIRST, THE PHYSICAL
0114 687 : TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION. SECOND, A
0114 688 : REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
0114 689 : NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXESWRITE_TODR ENTRY
0114 690 : WHICH WILL FABRICATE A NEW QUADWORD SYSTEM TIME.
0114 691
0114 692 : NOT NAUTILUS - FALL THROUGH TO WRITE_TODR
0114 693
0114 694 EXESWRITE_TODR:: ; SUBROUTINE ENTRY
0114 695
0114 696
0114 700
0114 701
0114 705
0114 706
0114 710
0114 711
1B 50 DA 0114 713 MTPR RO,#PR790$_TODR ; TODR IS A PROCESSOR REGISTER.
0117 715
0117 716
0117 721
05 0117 722 RSB
```

```
0118 724 .SBTTL EXESREGSAVE - SAVE CPU-SPECIFIC IPR'S
0118 725
0118 726 :↑ EXESREGSAVE - CALLED BY POWERFAIL TO SAVE CPU-SPECIFIC IPR'S ON
0118 727 : THE STACK
0118 728
0118 729 INPUTS: NONE
0118 730
0118 731 OUTPUTS:
0118 732
0118 733 RO DESTROYED
0118 734 OTHER GENERAL REGISTERS PRESERVED
0118 735 IPR'S SAVED ON THE STACK AS FOLLOWS:
0118 736
0118 737 11/780: 11/750: 11/730: 11/790: UVAX I:
0118 738
0118 739 0(SP) PME PME PME ACCS (none)
0118 740 4(SP) SBIMT TBDR CSWP
0118 741 8(SP) CADR PME
0118 742
0118 743 :-
0118 744
0118 745 .ENABL LSB
0118 746
01  BA 0118 747 EXESREGSAVE:: :SUBROUTINE ENTRY
0118 749 POPR #^M<R0> :CLEAR RETURN FROM STACK
011A 750
011A 751
011A 756
011A 757
011A 763
011A 764
011A 768
011A 769
011A 771 MFPR #PR790$_PME,-(SP) :SAVE PERFORMANCE MONITOR ENABLE
7E 00000042 7E 3D DB 011D 772 MFPR #PR790$_CSWP,-(SP) :SAVE CACHE STATE
7E 28 DB 0124 773 MFPR #PR790$_ACCS,-(SP) :SAVE FBOX STATE
00000042 8F 04 DA 0127 774 MTPR #CSWPSM_VAL,#PR790$_CSWP :SWEEP AND DISABLE CACHE
012E 776
60 17 012E 777 JMP (R0) :DONE, RETURN
0130 779
0130 783
0130 784 .DSABL LSB
```

```
0130 786 .SBTTL EXESREGRESTOR - RESTORE CPU-SPECIFIC IPR'S
0130 787
0130 788 EXESREGRESTOR - CALLED BY POWERFAIL RECOVERY TO RESTORE CPU-SPECIFIC
0130 789 IPR'S FROM THE STACK.
0130 790
0130 791 INPUTS:
0130 792
0130 793 R6 - TOP OF STACK
0130 794 STACK SET UP AS DEFINED IN OUTPUTS OF EXESREGSAVE.
0130 795
0130 796 OUTPUTS:
0130 797
0130 798 R0 DESTROYED
0130 799 OTHER GENERAL REGISTERS PRESERVED
0130 800 CPU-SPECIFIC IPR'S RESTORED FROM STACK
0130 801 R6 - ADDRESS OF 1ST CPU-INDEPENDENT SAVED IPR
0130 802
0130 803
0130 804
0130 805 .ENABL LSB
0130 806
0130 807 EXESREGRESTOR::
0130 809 POPR #*M<R0>
0132 810
0132 811
0132 816
0132 817
0132 823
0132 824
0132 828
0132 829
0132 831 BICL #*C<ACCSSM ENABLE>, (R6) ; ONLY WRITE FBOX ENABLE BIT
0139 832 MTPR (R6)+, #PR790$ ACCS ; RESTORE FBOX STATE
013C 833 BISL #CSWP$M INV, (R6) ; CAUSE CACHE SWEEP AND INVALIDATE
013F 834 MTPR (R6)+, #PR790$ CSWP ; SWEEP CACHE AND RESTORE ITS STATE
0146 835 MTPR (R6)+, #PR790$ PME ; RESTORE PERFORMANCE MONITOR ENABLE
0149 837
0149 838 JMP (R0) ; DONE, RETURN
0148 843
0148 844 .DSABL LSB
```

01 BA 0130 809

66 FFFF7FFF 8F CA 0132 831

28 86 DA 0139 832

66 08 CB 013C 833

00000042 8F 86 DA 013F 834

3D 86 DA 0146 835

60 17 0149 837

0148 843

0148 844



```
014B 846 .SBTTL EXESINIPROCREG - CPU-DEPENDENT INITIALIZATION OF IPR'S
014B 847 :+
014B 848 EXESINIPROCREG - PERFORM INITIALIZATION OF INTERVAL TIMER AND
014B 849 CPU-DEPENDENT REGISTERS. CALLED FROM INIT AND POWERFAIL.
014B 850
014B 851 INPUTS:
014B 852
014B 853 NONE
014B 854
014B 855 OUTPUTS:
014B 856
014B 857 NONE
014B 858
014B 859
014B 860 EXESINIPROCREG:: ; INIT PROCESSOR REGISTERS
014B 861 ; FOR 11/790:
014B 884
014B 885 :
014B 886 : Find all SBIA's and initialize their registers.
014B 887 :
014B 888 PUSHB #M<R0,R1,R2,R3> ; PRESERVE REGISTERS
014B 889 CLRL R2 ; INDEX INTO ABUS ARRAYS
014F 890 5$:
014F 891 CMPB #10790$C_SBIA,W*ABUS_TYPE[R2] ; IS THIS AN SBIA?
0155 892 BNEQ 8$ ; NO, KEEP LOOKING
0157 893 MOVL W*ABUS VA[R2],R1 ; GET VA OF SBIA REGISTER SPACE
015D 894 MOVL #SBIA$M_BEL,- ; CLEAR BUFFER ERROR LOCK BIT
0165 895 SBIA$M_SUMRY(R1) ; IN ERROR SUMMARY REGISTER
0165 896 MOVL #SBIA$M_CTO,- ; CLEAR CPU TIMEOUT ERROR
016D 897 SBIA$M_SBIERR(R1) ; IN SBI ERROR REGISTER
016D 898 MOVL #<SBIA$M_FLTLA!SBIA$M_FIE>,- ; CLEAR FAULT LATCH AND ENABLE
0175 899 SBIA$M_SBISTS(R1) ; FAULT INTERRUPTS IN SBI FLT/STS REG
0175 900 8$:
0175 901 AOBLS #4,R2,5$ ; LOOP THROUGH ALL ABUS ADAPTERS
0179 902 :
0179 903 : Enable CRD interrupts if requested.
0179 904 :
0179 905 : Due to a hardware bug, we must never run with CRD errors turned off
0179 906 :
0179 907 BBC S*#EXESV_CRDENABL,-
0179 908 @#EXESGL_FLAGS,10$ ; IF CLR, IGNORE CRD ERRORS
0179 909 MFPR #PR790$_MERG,R1 ; READ MEMORY ERROR REGISTER
0180 910 BBCC #MERGSV-INHCRD,R1,10$ ; CLEAR CRD INHIBIT (ENABLE INTERRUPTS)
0184 911 MTPR R1,#PR790$_MERG ; WRITE VALUE BACK TO THE REGISTER
0188 912 10$:
0188 913 :
0188 914 : For VENUS, the cache and FBOX are turned OFF at the beginning of booting.
0188 915 : Cache was turned on by calling the routine INISCACHE; turn the fbox on now.
0188 916 : We wait until now so that we can still boot if there are
0188 917 : severe problems with the cache or FBOX; MCHECK790's error handling
0188 918 : mechanisms are in place now.
0188 919 :
0188 920 MTPR #ACCS$M_ENABLE,- ; TURN ON FLOATING POINT ACCELERATION
0192 921 #PR790$_ACCS
0192 922 POPR #M<R0,R1,R2,R3> ; RESTORE REGISTERS
0194 923
0194 924 20$: BBS S*#EXESV_NOCLOCK,-
```

H 12  
 - ERROR SUBROUTINES FOR VAX 11/790 16-SEP-1984 00:59:29 VAX/VMS Macro V04-00  
 EXESINIPROCREG - CPU-DEPENDENT INITIALIZ 13-SEP-1984 15:49:22 [SYSLOA.SRC]ERRSUB.MAR;5

Page 16  
(11)

	0E	00000000'9F		0196	926		#EXESGL_FLAGS,30\$	; BRANCH IF NOT USING CLOCK
				019C	927			
19	FFFFD8F0	8F	DA	019C	929	MTPR	#-<10*1000>,S^#PR790\$_NICK	; LOAD NEXT INTERVAL REGISTER
				01A3	931			
				01A3	935			
				01A3	939			
				01A3	943			
18	800000D1	8F	DA	01A3	944	MTPR	#^X800000D1,S^#PRS_ICCS	; CLEAR ERROR AND START CLOCK
			05	01AA	945	RSB		; AND RETURN
				01AB	946			
				01AB	962			

```
01AB 965 .SBTTL INISCACHE
01AB 966 :++
01AB 967 : INISCACHE - ON 11/790, INITIALIZE AND TURN ON CACHE
01AB 968 : - THIS ROUTINE NOT USED FOR OTHER PROCESSORS
01AB 969 :
01AB 970 : THIS ROUTINE IS CALLED TO INITIALIZE AND TURN ON THE CACHE. FOR THE VENUS
01AB 971 : PROCESSOR, CACHE IS DISABLED WHILE WE ARE BOOTING. IT IS ONLY ENABLED
01AB 972 : AFTER THE VENUS MACHINE CHECK HANDLER IS CONNECTED, SO THAT RECOVERABLE
01AB 973 : CACHE ERRORS DO NOT PREVENT THE SYSTEM FROM BOOTING.
01AB 974 :
01AB 975 : HOWEVER, CACHE MUST BE ENABLED BEFORE THE EXESGL UBDELAY AND EXESGL TENUSEC
01AB 976 : CELLS (USED BY DRIVERS WHEN THEY USE THE TIMEDWAIT MACRO) ARE CALIBRATED.
01AB 977 :
01AB 978 :--
01AB 979 INISCACHE::
01AB 980 MTPR #<CSWPSM_COENA+CSWPSM_C1ENA+CSWPSM_INV>, -
01B2 981 #PR790$_CSWP ; TURN ON BOTH HALVES OF CACHE
05 01B2 982 RSB
```

00000042 8F 0B DA



```
01B3 985 .SBTTL SYSLSCLR SBIA
01B3 986
01B3 987 ++
01B3 988 SYSLSCLR SBIA - ON 11/790, CLEAR SBIA ERROR REGISTERS
01B3 989 - ON 11/780, 11/750, 11/730, AND MICRO-VAX I, THIS IS A NOP
01B3 990
01B3 991 THIS ROUTINE IS CALLED TO CLEAR OUT SBIA ERROR BITS AFTER A MACHINE CHECK
01B3 992 OCCURS (WHEN MACHINE CHECK IS HANDLED LOCALLY).
01B3 993
01B3 994 THIS ROUTINE SHOULD BE CALLED AT IPL 31.
01B3 995
01B3 996 INPUTS:
01B3 997 ABUS_TYPE - AN ARRAY TYPE CODES; IDENTIFIES EACH ADAPTER ON THE
01B3 998 ABUS.
01B3 999 ABUS_VA - AN ARRAY OF ADAPTER SPACE VA'S FOR EACH ADAPTER
01B3 1000 ON THE ABUS.
01B3 1001
01B3 1002 OUTPUTS:
01B3 1003 SBI ERROR BITS ARE CLEARED FOR EACH SBIA ON THE ABUS.
01B3 1004 ALL REGISTERS PRESERVED.
01B3 1005 ++
01B3 1006 SYSLSCLR SBIA::
01B3 1007 PUSH R1, R2 ; SAVE SOME REGISTERS
01B3 1008 CLRL R2 ; INDEX INTO ABUS ARRAYS
01B3 1009 10$:
01B3 1010 CMPB #10790$C_SBIA, W*ABUS_TYPE[R2] ; IS THIS AN SBIA?
01B3 1011 BNEQ 20$ ; NO, KEEP LOOKING
01B3 1012 MOVL W*ABUS_VA[R2], R1 ; GET VA OF SBIA REGISTER SPACE
01B3 1013 MOVL SBIA$SL_SUMRY(R1), - ; CLEAR ERRORS
01B3 1014 SBIA$SL_SUMRY(R1) ; IN ERROR SUMMARY REGISTER
01B3 1015 MOVL SBIA$SL_SBIERR(R1), - ; CLEAR ERRORS
01B3 1016 SBIA$SL_SBIERR(R1) ; IN SBI ERROR REGISTER
01B3 1017 MOVL SBIA$SL_SBISTS(R1), - ; CLEAR ERRORS
01B3 1018 SBIA$SL_SBISTS(R1) ; IN SBI FAULT/STATUS REGISTER
01B3 1019 20$:
01B3 1020 AOBLS #4, R2, 10$ ; LOOP THROUGH ALL ABUS ADAPTERS
01B3 1021 POP R1, R2 ; RESTORE REGISTERS
01B3 1022 RSB ; AND RETURN
01B3 1023
```

06	BB	01B3	1007
52	D4	01B5	1008
		01B7	1009
0000'CF42	01	01B7	1010
	15	01BD	1011
51 0000'CF42	D0	01BF	1012
08 A1 08 A1	D0	01C5	1013
		01CA	1014
34 A1 34 A1	D0	01CA	1015
		01CF	1016
3C A1 3C A1	D0	01CF	1017
		01D4	1018
		01D4	1019
DF 52 04	F2	01D4	1020
	06	01D8	1021
	05	01DA	1023

```
01DB 1025 .SBTTL EXESTEST_CSR
01DB 1026
01DB 1031 *+ EXESTEST_CSR - TEST A UNIBUS CONTROLLER CSR FOR EXISTENCE
01DB 1033
01DB 1034 THIS TEST IS CPU-DEPENDENT. THE FOLLOWING CPU'S ARE SUPPORTED:
01DB 1035
01DB 1036 11/780 -TEST CSR AND CHECK RESULT IN THE UBA STATUS REGISTER.
01DB 1037 11/750 -NON-EXISTENT CSR IS REPORTED VIA MACHINE CHECK AS A
01DB 1038 NON-EXISTENT MEMORY REFERENCE. CONNECT A TEMPORARY
01DB 1039 MACHINE CHECK HANDLER, TEST THE CSR, AND RESTORE THE
01DB 1040 ORIGINAL MACHINE CHECK HANDLER.
01DB 1041 11/730 -ACTION IS THE SAME AS FOR THE 11/750.
01DB 1042 11/790 -ACTION IS THE SAME AS FOR THE 11/780.
01DB 1043 MICRO-VAX I -ACTION IS SAME AS FOR THE 11/750.
01DB 1044
01DB 1045 THIS SUBROUTINE SHOULD BE CALLED VIA BRANCH OR JUMP TO SUBROUTINE AT IPL 31.
01DB 1046
01DB 1047 INPUTS:
01DB 1048
01DB 1049 R0 = CSR ADDRESS
01DB 1050 R6 = ADAPTER CONFIGURATION REGISTER ADDRESS
01DB 1051
01DB 1052 OUTPUTS:
01DB 1053
01DB 1054 R0 LOW BIT SET/CLEAR FOR EXISTENT/NONEX CSR
01DB 1055 OTHER REGISTERS PRESERVED.
01DB 1056 :-
01DB 1057
01DB 1058 .ENABL LSB
01DB 1059
01DB 1060 EXESTEST_CSR:: ;SUBROUTINE ENTRY
01DB 1061
06 BB 01DB 1062 PUSHF #M<R1,R2> ;SAVE REGISTERS
01DD 1063
01DD 1064
01DD 1065
01DD 1066 : This next line of code is present so that this routine continues to function
01DD 1067 : correctly when the UNIBUS adapter is powered down. Moving 0 into the UBA
01DD 1068 : Status Register has no effect when addressing the actual adapter register,
01DD 1069 : and clears out any garbage bits in memory when UNIBUS space is re-mapped to
01DD 1070 : the "black hole" page.
01DD 1071 :
01DD 1072
01DD 1073 MOVL #0,UBASL_SR(R6) ;WHEN UBA IS REMAPPED
01E1 1074 MOVL G^EXESGL_SCB,R1 ;GET SCB ADDRESS
01E8 1075 PUSHF 4(R1) ;SAVE CURRENT MCHK HANDLER ADDR
01EB 1076 MOVL SP,R2 ;MARK CURRENT STACK POSITION
01EE 1077 MOVAL B^MCHK_790,4(R1) ;CONNECT TEMP 11/790 MCHK HANDLER
01F3 1078 TSTW (R0) ;ATTEMPT TO READ CSR
01F5 1079 MOVL UBASL_SR(R6),UBASL_SR(R6) ;CLEAR AND CHECK FOR ERROR
01FA 1080 BNEQ NONEX_DEV ;BRANCH IF ERROR
01FC 1081 MOVZBL #SS$_NORMAL,R0 ;SET STATUS TO SUCCESS
01FF 1082 BRB TEST_DONE ;JOIN COMMON EXIT
0201 1083
0201 1084
0201 1085
0201 1086
0201 1087
0201 1088
0201 1103
0201 1105 :
0201 1106 : TEMPORARY CSR TEST MACHINE CHECK HANDLER FOR THE 11/790:
0201 1107 :
0201 1108 .ALIGN LONG
```

51 08 A6 00 D0 01DD 1072  
00000000 GF D0 01E1 1073  
04 A1 DD 01E8 1074  
52 5E D0 01EB 1075  
04 A1 04 AF DE 01EE 1076  
60 B5 01F3 1077  
08 A6 08 A6 D0 01F5 1078  
28 12 01FA 1079  
50 01 9A 01FC 1080  
25 11 01FF 1081

```
50 0000004A BF DB 0204 1109 MCHK_790:
    00 50 05 E5 0204 1110 MFPR #PR790$EHSR,R0 ;GET ERROR HANDLING STATUS REG
0000004A BF 50 DA 020B 1111 BBCC #EHSR$V-VMS,R0,10$ ;CLEAR VMS ENTERED BIT
    FF 9A 30 020F 1112 10$: MTPR R0,#PR790$EHSR ;WRITE BACK TO REGISTER
    50 3C AE D0 0216 1113 BSBW SYSL$CLRSBIA ;CLEAR SBIA ERROR BITS
    SE 52 D0 0219 1114 MOVL MCF790$L_MSTAT2(SP),R0 ;PICK UP MEMORY STATUS REGISTER
    D8 50 02 E1 021D 1115 MOVL R2,SP ;CLEAR MACHINE CHECK FRAME OFF STACK
    0220 1116 BBC #MSTAT2$V_IOBUFF,R0,OK ;IF NOT NXM THEN SOMETHING'S THERE
    0224 1118
    50 D4 0224 1188 NONEX_DEV: ;
    0224 1189 CLRL R0 ;SET STATUS TO FAILURE
    04 A1 8ED0 0226 1190 TEST_DONE: ;
    0226 1191 POPL 4(R1) ;RESTORE SYSTEM MCHECK HANDLER
    06 BA 022A 1192 TEST_DONE_2: ;
    05 05 022A 1193 POPR #^M<R1,R2> ;RESTORE REGISTERS
    022C 1194 RSB ;RETURN RESULT TO CALLER
    022D 1195 .DISABLE LSB
```



```

022D 1197 .SBTTL ADPLINK - LINK ADAPTER CONTROL BLOCK INTO ADP LIST
022D 1198 :+
022D 1199 : ADPLINK LINKS THE ADAPTER CONTROL BLOCK TO THE END OF THE ADP LIST
022D 1200 :
022D 1201 : INPUT:
022D 1202 : R2 - ADDRESS OF NEW ADP
022D 1203 : OUTPUTS:
022D 1204 : ADP IS LINK TO THE END OF THE ADPLIST LOCATED BY IOC$GL_ADPLIST.
022D 1205 : R0,R1 destroyed.
022D 1206 :-
022D 1207 :
022D 1208 ADPLINK::
50  FFFFFFFC'9F 9E 022D 1209 MOVAB @#<IOC$GL_ADPLIST-ADP$LINK>,R0
    51  04 A0 D0 0234 1210 : START OF LIST
    50  05 13 0234 1211 10$: MOVL ADP$LINK(R0),R1 : FLINK TO FIRST ENTRY
    50  51 D0 0238 1212 : BEQL 20$ : AT END
    04 A0 52 D0 023A 1213 : MOVL R1,R0 : TRY AGAIN
    11 023D 1214 : BRB 10$ :
    D0 023F 1215 20$: MOVL R2,ADP$LINK(R0) : CHAIN NEW ADP TO END OF LIST
    05 0243 1216 : RSB : AND RETURN
    0244 1217 :
    0244 1218 .END

```

ERRSUB790  
Symbol table

- ERROR SUBROUTINES FOR VAX 11/790 N 12

16-SEP-1984 00:59:29 VAX/VMS Macro V04-00  
13-SEP-1984 15:49:22 [SYSLOA.SRC]ERRSUB.MAR;5

Page 22  
(15)

ABUS_TYPE	*****	X	03	MCF790\$M_MSTAT2	= 0000003C		
ABUS_VA	*****	X	03	MCHK_790	= 00000204	R	03
ACCSM_ENABLE	= 00008000			MERG\$V_INHCRD	= 0000000A		
ADP\$M_CSR	= 00000000			MSTAT2\$V_IOBUFF	= 00000002		
ADP\$M_LINK	= 00000004			NDT\$-CI	= 00000038		
ADP\$M_ADPTYPE	= 0000000E			NDT\$-MB	= 00000020		
ADPLINK	0000022D	RG	03	NONEX_DEV	00000224	R	03
ADP_TBL_DWN	0000009E	R	03	OK	000001FC	R	03
ADP_TBL_UP	00000086	R	03	PR\$-ICCS	= 00000018		
BQO\$M_UMR_DIS	= 00000024			PR\$-SID_TYP730	= 00000003		
BQO\$M_VERSION	= 00000010			PR\$-SID_TYP750	= 00000002		
BTD\$M_CONSOLE	= 00000040			PR\$-SID_TYP780	= 00000001		
C750_LIKE	= 00000000			PR\$-SID_TYP790	= 00000004		
C780_LIKE	= 00000001			PR\$-SID_TYPUV1	= 00000007		
CIS\$SHUTDOWN	*****	X	03	PR790\$-ACCS	= 00000028		
CPU_TYPE	= 00000004			PR790\$-CSWP	= 00000042		
CSWPSM_COENA	= 00000001			PR790\$-EHSR	= 0000004A		
CSWPSM_C1ENA	= 00000002			PR790\$-ICR	= 0000001A		
CSWPSM_INV	= 00000008			PR790\$-MERC	= 00000047		
CSWPSM_VAL	= 00000004			PR790\$-NICR	= 00000019		
EHSR\$V-VMS	= 00000005			PR790\$-PME	= 0000003D		
EXE\$DUMPCPUREG	000000CE	RG	03	PR790\$-TODR	= 0000001B		
EXE\$EXTRA1	00000000	RG	01	RPB\$B_DEVTYPE	= 00000066		
EXE\$EXTRA10	00000000	RG	01	RPB\$M-ADPVIR	= 00000060		
EXE\$EXTRA2	00000000	RG	01	RPB\$M-BOOTR1	= 00000020		
EXE\$EXTRA3	00000000	RG	01	RPB\$M-IOVEC	= 00000034		
EXE\$EXTRA4	00000000	RG	01	RPB\$M-ABUS	= 00000002		
EXE\$EXTRA5	00000000	RG	01	RPB\$M-ABUS	= 00000004		
EXE\$EXTRA6	00000000	RG	01	RPB\$M-BOOTNDT	= 000000A1		
EXE\$EXTRA7	00000000	RG	01	SBIASL_CR	00000000		
EXE\$EXTRA8	00000000	RG	01	SBIASL_CSR	00000004		
EXE\$EXTRA9	00000000	RG	01	SBIASL-DIAGNOS	0000000C		
EXE\$GL_FLAGS	*****	X	03	SBIASL-DMAACA	00000018		
EXE\$GL_RPB	*****	X	03	SBIASL-DMAAID	0000001C		
EXE\$GL_SCB	*****	X	03	SBIASL-DMABCA	00000020		
EXE\$INIBOOTADP	00000000	RG	03	SBIASL-DMABID	00000024		
EXE\$INIPROCREG	0000014B	RG	03	SBIASL-DMACCA	00000028		
EXE\$READP_TODR	00000110	RG	03	SBIASL-DMACID	0000002C		
EXE\$READ_TODR	00000110	RG	03	SBIASL-DMAICA	00000010		
EXE\$REGRESTOR	00000130	RG	03	SBIASL-DMAID	00000014		
EXE\$REGSAVE	00000118	RG	03	SBIASL-MAINT	00000044		
EXE\$SHUTDOWNADP	00000077	RG	03	SBIASL-SBIERR	00000034		
EXE\$STARTUPADP	0000006F	RG	03	SBIASL-SBIQC	0000004C		
EXE\$TEST_CSR	000001DB	RG	03	SBIASL-SBISILO	00000030		
EXE\$V_NOCLOCK	*****		03	SBIASL-SBISTS	0000003C		
EXE\$WRITEP_TODR	00000114	RG	03	SBIASL-SILOCMP	00000040		
EXE\$WRITE_TODR	00000114	RG	03	SBIASL-SUMRY	00000008		
INISCACHE	000001AB	RG	03	SBIASL-TMOADDRS	00000038		
INI_UBADP	00000038	R	03	SBIASL-UNJAM	00000048		
IO790\$C_SBIA	= 00000001			SBIASL-BEL	= 00800000		
IOC\$GL_ADPLIST	*****	X	03	SBIASL-CTO	= 00001000		
MASINITIAL	*****	X	03	SBIASL-FIE	= 00040000		
MBASINITIAL	*****	X	03	SBIASL-FLTLA	= 00080000		
MBASL_CR	= 00000004			SS\$ NORMAL	= 00000001		
MBASL_SR	= 00000008			SYS\$CLRSBIA	000001B3	RG	03
MBASH_CR_ABORT	= 00000002			TEST_DONE	00000226	R	03
MBASH_CR_INIT	= 00000001			TEST_DONE_2	0000022A	R	03



ERRSUB790  
Symbol table

- ERROR SUBROUTINES FOR VAX 11/790<sup>B 13</sup>

16-SEP-1984 00:59:29 VAX/VMS Macro V04-00  
13-SEP-1984 15:49:22 [SYSLOA.SRC]ERRSUB.MAR;5

Page 23  
(15)

UBAS\$INITIAL           \*\*\*\*\* X 03  
UBAS\$L\_CR            = 00000004  
UBAS\$L\_CSR           = 00000000  
UBAS\$L\_MAP           = 00000800  
UBAS\$L\_SR            = 00000008  
UBAS\$M\_CR\_INIT       = 00000001  
UBAS\$M\_CSR\_UBIC      = 00010000

-----  
! Psect synopsis !  
-----

PSECT name	Allocation	PSECT No.	Attributes
. ABS	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK	00000001 ( 1.)	01 ( 1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$AB\$\$	00000050 ( 80.)	02 ( 2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SYSLOA	00000244 ( 580.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.03	00:00:02.27
Command processing	116	00:00:00.52	00:00:05.79
Pass 1	458	00:00:11.29	00:00:39.07
Symbol table sort	0	00:00:01.76	00:00:06.00
Pass 2	152	00:00:02.53	00:00:28.95
Symbol table output	15	00:00:00.07	00:00:00.08
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	777	00:00:16.22	00:01:22.18

The working set limit was 1650 pages.  
102996 bytes (202 pages) of virtual memory were used to buffer the intermediate code.  
There were 90 pages of symbol table space allocated to hold 1658 non-local and 22 local symbols.  
1222 source lines were read in Pass 1, producing 16 object records in Pass 2.  
28 pages of virtual memory were used to define 27 macros.

-----  
! Macro library statistics !  
-----

Macro library name	Macros defined
_\$255\$DUA28:[SYSLOA.OBJ]790DEF.MLB;1	7
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	11
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	24

1777 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:ERRSUB790/OBJ=OBJ\$:ERRSUB790 MSRC\$:CPUSW790/UPDATE=(ENH\$:CPUSW790)+MSRC\$:ERRSUB/UPDATE=(ENH\$:ERRSUB)+EXECML\$/LIB+LIB\$



0395

AH-BT13A-SE  
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY